REMARKS

In view of the following remarks, reconsideration and allowance of this application are requested. Claims 1-11, 13-16 and 31-50, and 53-62 are pending, with claims 1 and 10 being independent. Claims 12 and 51-52 have been cancelled.

35 U.S.C. § 103(a) Caire et al./Berry et al. Rejection

Claims 1, 4, 9-11, 13-16, and 31-50 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,663,962 ("Caire") in view of U.S. Patent No. 5,692,205 ("Berry"). Claims 4, 9, and 31-40 depend from claim 1. Claims 11-16 and 41-50 depend from claim 10. This rejection, insofar as it pertains to the presently pending claims, is respectfully traversed.

Independent claims 1 and 10 are directed to producing a streaming multimedia document and recite a "presentation being arranged so as to be independent of a bandwidth of a communications channel used to send the multimedia document to the user and to incrementally render the objects to the user according to the organization." Neither Caire nor Berry, alone or in combination, teach or suggest at least this feature of independent claims 1 and 10.

Caire is directed to a method for multiplexing streams of audio-visual signals coded according to the MPEG1 standard. A time-division multiplexing process is used to construct a stream of packets, where each packet contains a single type of data. See Caire at col. 1, l. 65 to col. 2, l. 2. The method involves building a single multiplexed stream from multiple individual elementary streams by continuously deciding, based upon the urgency that data is needed by a demultiplexing buffer, from which elementary stream to select data and form a packet so that the buffer does not become empty. See Caire at abstract; col. 1, ll. 37-45, 52-64; col. 2, ll. 3-9, 45-59; col. 4, l. 66 to col. 5, l. 7; Figs 1, 1A, 7A, 7B. The single multiplexed stream is transmitted to a remote demultiplexer, which divides the multiplexed stream into the component individual elementary steams and stores the streams in a demultiplexing buffer before being supplied to a decoder. See Caire at col. 1, ll. 37-45, 52-64; col. 2, ll. 3-9, 45-59; col. 4, ll. 54-65. At a given instant in time, Caire evaluates the amount of data currently contained in a demultiplexing buffer

based on the difference between the amount of data that has already arrived and the amount of data already extracted from the demultiplexing buffer. See Caire at col. 2, l. 59 to col. 3, l. 2. Caire then calculates a "relax" parameter linked to the buffer occupancy level and representative of how urgently the buffer needs to receive data to avoid an underflow condition, identifies the individual elementary stream with the minimum relax parameter, and constructs a data packet to transmit based on the individual elementary stream having the minimum relax parameter. See Caire at col. 3, ll. 3-40; col. 6. ll. 48-67. Thus, Caire "aims at preventing underflow conditions," and also teaches how to avoid buffer overflow conditions by constantly adjusting the individual elementary stream from which data is selected in order to form the multiplexed stream. Caire at col. 6, ll. 14-15. In other words, Caire multiplexes data on the fly, making adjustments to the arrangement of a presentation based on the communications channel and how full the demultiplexing buffers are at each instance of time.

As such, Caire fails to describe or suggest a presentation being arranged so as to be independent of a bandwidth of a communications channel used to send the multimedia document to the user and to incrementally render the objects to the user according to the organization, as recited in claims 1 and 10.

Berry is directed to a method and system for integration of multimedia presentations within an object oriented user interface, which provides for encapsulating multimedia data within an object. See Berry at col. 2, ll. 41-45. Berry does not require separate audio or video objects to be manipulated by the user. See Berry at col. 2, ll. 41-45. Nevertheless, like Caire, Berry fails to describe or suggest a presentation being arranged so as to be independent of a bandwidth of a communications channel used to send the multimedia document to the user and to incrementally render the objects to the user according to the organization, as recited in claims 1 and 10.

Independent claims 1 and 10 thus are allowable for at least this reason. Claims 4, 9, 11, 13-16, and 31-50 are allowable by virtue of their dependency, as well as on their own merits.

Accordingly, reconsideration and withdrawal of this rejection are respectfully requested.

35 U.S.C. § 103(a) Caire/Berry/Ando Rejection

Claims 2, 3, 7, and 8 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Caire, and Berry, and further in view of U.S. Patent No. 5,600,826 to Ando ("Ando"). Claims 2,

3, 7, and 8 depend from claim 1. This rejection, insofar as it pertains to the independent claims, is respectfully traversed.

Ando is directed to a structured data processor for converting between sequential and tree structured data, including a structured data treating unit for editing data. See Ando at col. 4, ll. 25-43; col. 6, ll. 44-47. Ando fails to remedy the deficiencies of Caire and Berry with respect to independent claim 1. Claims 2, 3, 7, and 8 thus are allowable by virtue of their dependency, as well as on their own merits.

Accordingly, reconsideration and withdrawal of this rejection are respectfully requested.

35 U.S.C. § 103(a) Caire/Berry/Johnson Rejection

Claims 5 and 6 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Caire and Berry, and further in view of U.S. Patent No. 5,892,847 to Johnson ("Johnson"). Claims 5 and 6 depend from claim 1. This rejection, insofar as it pertains to the independent claims, is respectfully traversed.

Johnson is directed to a method and apparatus for compressing images, including an encoder that that created a file format that layers the compressed image. See Johnson at col. 4, ll. 30-49. Johnson fails to remedy the deficiencies of Caire and Berry with respect to independent claim 1. Claims 5 and 6 thus are allowable by virtue of their dependency, as well as on their own merits.

Accordingly, reconsideration and withdrawal of this rejection are respectfully requested.

Additional Claims 53-62

Claims 53-57 depend from claim 1 and claims 58-62 depend from claim 10. Thus, claims 53-62 are allowable by virtue of their dependency, as well as on their own merits.

Attached is a marked-up version of the changes being made by the current amendment.

Applicant submits that all of the claims are in condition for allowance. Enclosed is a $126.00 check for excess claim fees. Please apply any other charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: July 1, 2002

Scott R. Boalick
Reg. No. 42,337

Fish & Richardson P.C.
1425 K Street, N.W.
11th Floor
Washington, DC 20005-3500
Telephone: (202) 783-5070
Facsimile: (202) 783-2331

Version with markings to show changes made

In the claims:

Claims 1-6, 8-11, 13-14, 31-37, 39-47, and 49-50 have been amended as follows:

1. A method for producing a **[hierarchical data file for a]** streaming multimedia document **[having a plurality of object files, the data file having different file formats encapsulated within the data file]**, the method comprising:

encapsulating **[in]** within a single file **[multimedia document a first file support]** at least two objects, each object including data for the object and data defining a relationship of the object within a multimedia document **[information in a first file format]**; and

[supporting the first file support object by the multimedia document;]

[encapsulating in the multimedia document a second file support object including information in a second file format;

supporting the second file support object by the multimedia document;]

[encapsulating in the multimedia document choreographing information for allowing] defining a presentation of each object to a user according to an organization of the file, the organization being controlled by a document author, **[to define the timing at which the first file support object and the second file support object are retrieved by a user, the choreographing information comprising data slices from the first file support object and the second file support object multiplexed so as]** the presentation being arranged so as to be independent of a bandwidth of a communications channel used to send the multimedia document to the user and to incrementally render the **[first file support]** objects **[and the second file support object]** to the user according to the organization **[at substantially the same time]**.

2. The method of claim 1, further comprising changing at least one of the objects in the **[data]** file.

3. The method of claim 1, further comprising adding at least one object to the **[data]** file.

4. The method of claim 1 wherein the **[data]** file is displayed in a window on a computer display, the method further comprising:

creating an exclusionary area within the window; and

locating an object within the exclusionary area, the object being selected from a group of **[data]** objects including a framed image, a slide show, framed text, sound data, a separator, or a hyperlink.

5. The method of claim 1 wherein the **[data]** file includes splash image data defining a splash image, the method further comprising locating the splash image data within the **[data]** file such that the splash image is displayed on a computer display as the splash image data is received by a receiver coupled to the computer display.

6. The method of claim 5, further comprising locating update splash data that further defines the splash image within the **[data]** file such that the splash image is updated on the computer display as the receiver receives the update splash data.

8. The method of claim 1, further comprising compressing the **[information in each]** <u>data for the object in at least one</u> object.

9. The method of claim 1 wherein the data file is downloaded by a receiving computer, the method further comprising:

creating an unknown object in the **[data]** file; and

locating player data within the unknown object defining a player that plays the unknown object.

10. A computer system having a memory storing a **[hierarchical data]** file structure **[that encapsulates a plurality of different file formats to form a streaming multimedia**

document having a plurality of object files, the multimedia document being capable of being displayed on a display of a computer system], the **[data]** file structure comprising:

[a document including information for controlling the display;]

at least two objects **[a first support object including information in a first file format, the first support object being]** encapsulated within a single file, each object including data for the object and data defining a relationship of the object within a multimedia document **[in the document and being capable of supporting a plurality of first lower objects, each first lower object being a lower level object than the first support object in the hierarchical data file structure;**

a second support object including information in a second file format, the second support object being encapsulated in the document and being capable of supporting a plurality of second lower objects, each second lower object being a lower level object than the second support object in the hierarchical data file structure]; and

an organization of the file according to which a presentation of each object to a user may be defined, the organization being controlled by **[choreographing information for allowing]** a document author, **[to define the timing at which the first file support object and the second file support object are retrieved by a user, the choreographing information comprising data slices from the first file support object and the second file support object multiplexed so as]** the presentation being arranged so as to be independent of a bandwidth of a communications channel used to send the multimedia document to the user and to incrementally render the **[first file support]** objects **[and the second file support object]** to the user according to the organization **[at substantially the same time]**.

11. The computer system of claim 10 wherein at least one object comprises one of **[the first file format is selected from a group of file formats including]** a textual file format, an image file format, and a sound file format**[; and wherein the second file format is selected from a group of file formats including a textual file format, an image file format, and a sound file format]**.

13. The computer system of claim 10 wherein **[each]** two or more objects **[has a plurality of]** have at least one common attribute**[s]**, including at least one of a command for perception of the object, an ability to pass and receive a message, and an ability to supply and retrieve the data embodied in the object.

14. The computer system of claim 10 wherein **[each]** at least one object is a generic element of the hierarchical data file structure, such that any combination of objects can be grouped together to form a part of the multimedia document.

31. The method of claim 56 **[1]**, wherein the choreography**[ing]** information further comprises:

a header;

an object archive for storing information about **[the plurality of]** one or more objects **[files]**, the object archive including information about the relationship of the **[level of each]** object file with the **[hierarchy]** document; and

a multiplex section including data for **[each of]** the objects **[files of]** in the document.

32. The method of claim 31, wherein the objects **[files]** in the multiplex section are each played by a player as the multiplexed object **[file]** is received by a receiver.

33. The method of claim 31, wherein the data for the objects **[files]** is interleaved in the multiplex section.

34. The method of claim 31, wherein the object archive includes data defining a geometry of the document.

35. The method of claim 31, wherein **[each of the]** one or more objects **[files]** is defined by at least one data slice; and wherein the multiplex section further includes:

an object number counter indicating the number of objects **[files]**;

a plurality of object descriptions, each object description describing a corresponding one of the objects **[files]**; and

a choreography group providing information about a first group of objects **[files]**.

36. The method of claim 35, wherein the choreography group further comprises:

a group object counter indicating the number of objects **[files]** in the choreography group;

size and type data for each object **[file]**;

header data; and

the data slices of the objects **[files]** interleaved together.

37. The method of claim 35, wherein the choreography group includes data slices of the objects **[files]** interleaved in a predetermined manner.

39. The method of claim 35, further comprising placing one or more **[locating a plurality of]** slice size data blocks before one or more of the interleaved data slices, each slice size data block corresponding to **[one of the]** a data slice**[s]** and providing a size of the corresponding data slice.

40. The method of claim 31, further comprising a non-multiplex section following the multiplex section, the non-multiplex section including **[a plurality of]** one or more separate objects **[files]** that are not played by a player as the separate object files are received by a receiver.

41. The computer system of claim 61 **[10]**, wherein the choreography**[ing]** information further comprises:

a header;

an object archive for storing information about **[the plurality of]** one or more objects **[files]**, the object archive including information about the relationship of the **[level of each]** object file with the **[hierarchy]** document; and

a multiplex section including data for **[each of]** the objects **[files of]** in the document.

42. The computer system of claim 41, wherein the objects **[files]** in the multiplex section are each played by a player as the multiplexed object **[file]** is received by a receiver.

43. The computer system of claim 41, wherein the data for the objects **[files]** is interleaved in the multiplex section.

44. The computer system of claim 41, wherein the object archive includes data defining a geometry of the document.

45. The computer system of claim 41, wherein **[each of the]** one or more objects **[files]** is defined by at least one data slice**[;]** and wherein the multiplex section further includes:

an object number counter indicating the number of objects **[files]**;

a plurality of object descriptions, each object description describing a corresponding one of the objects **[files]**; and

a choreography group providing information about a first group of objects **[files]**.

46. The computer system of claim 45, wherein the choreography group further comprises:

a group object counter indicating the number of objects **[files]** in the choreography group;

size and type data for each object **[file]**;

header data; and

the data slices of the objects **[files]** interleaved together.

47. The computer system of claim 45, wherein the choreography group includes data slices of the objects **[files]** interleaved in a predetermined manner.

49. The computer system of claim 45, further comprising placing one or more **[locating a plurality of]** slice size data blocks before one or more of the interleaved data slices, each slice size data block corresponding to **[one of the]** a data slice**[s]** and providing a size of the corresponding data slice.

50. The computer system of claim 41, further comprising a non-multiplex section following the multiplex section, the non-multiplex section including **[a plurality of]** one or more separate objects **[files]** that are not played by a player as the separate object files are received by a receiver.